REMARKS

Claim 1, and 3-20 stands rejected under 35 USC §103 as being unpatentable over Henrion et al, U.S. patent 6,469,982 in view of Snyder II et al., U.S. patent 6,888,830 and further in view of Bass et al., U.S. patent 6,804,249. Claim 2 stands rejected under 35 USC §103 as being unpatentable over Henrion et al, U.S. patent 6,469,982 in view of Snyder II et al., U.S. patent 6,888,830 and further in view of Bass et al., U.S. patent 6,804,249 and further in view of Holt et al., U.S. patent 5,790,545.

Claims 1, 4, 11, 12, 17, 18, 19, and 20 have been amended to more clearly state the invention. Reconsideration and allowance of each of the pending claims 1-20, as amended, is respectfully requested.

Henrion et al, U.S. patent 6,469,982 discloses a method of sharing available bandwidth on a common link in a communication network among a plurality of data flows which are transmitted via the common link. The method is used by a processor and includes sharing reserved bandwidth included in the available bandwidth among the plurality of data flows, and sharing unreserved bandwidth among the plurality of data flows according to a respective unreserved data packet share which is associated to each one of the plurality of data flows. The unreserved bandwidth is included in the available bandwidth in excess of the reserved bandwidth. The sharing of the unreserved bandwidth includes associating to one of the plurality of data flows a respective adaptable administrative weight and determining the respective unreserved data packet share which is associated to the one data flow as a function of its

respective adaptable administrative weight. At column 11, lines 5-35 state:

The second processor P2 includes a second determiner DET2 in order to determine the respective unreserved data packet shares in function of these adaptable administrative weights. It has to be remarked that the unreserved bandwidth is in this particular embodiment shared among the active data flows i.e. the data flows having at least one data packet in its associated buffer queue of the buffer BUF. How the second determiner DET2 shares the unreserved bandwidth among the active data flows will be explained in a further paragraph.

The second processor P2 includes also, a calculator in order to make some necessary calculations and a reader and a writer in order to interact with the memory MEM and the controler CTRL, however as mentioned above, these functional blocks are not shown in the FIGURE and are not described in further detail.

The processor P includes also a third determiner DET3 in order to determine a status of the data flows i.e. active data flow or not active data flow. In order to execute this function the third determiner DET controls at the buffer BUF for each buffer queue e.g. Q2 the eventual presence of a data packet. In the event when at least one data packet is present in a data queue the status of the associated data flow is confirmed to be "yes". In the event when no data packet is present in a data queue, the status of the associated data flow is confirmed to be "no". The status of each data flow is provided by the third determiner DET3 to the memory MEM described hereafter and is stored in this memory MEM for each data flow by means of a third variable called status variable ACT.

Snyder II et al., U.S. patent 6,888,830 discloses an integrated circuit processes a communication packet and comprises a core processor and scheduling circuitry. The core processor executes a software application that directs the core processor to process the communication packet. The scheduling circuitry retrieves first scheduling parameters cached in a context buffer for the packet and executes a first algorithm based on the first scheduling parameters to schedule subsequent transmission of the communication packet. At column 9, starting at line 52 states:

External buffers 528 use a linked list mechanism to store communication packets externally to integrated circuit 100. Pointer stack 527 is a cache of pointers to free external buffers 528 that is initially built by core processor 104. Pointer cache 523 stores pointers that were transferred from pointer stack 527 and correspond to external buffers 528. Sets of pointers may be periodically exchanged between pointer stack 527 and pointer cache 523. Typically, the exchange from stack 527 to cache 523 operates

on a first-in/first-out basis.

In operation, core processor 104 writes pointers to free external buffers 528 to pointer stack 527 in SDRAM 526. Through SDRAM interface 522, control logic 524 transfers a subset of these pointers to pointer cache 523. When an entity, such as core processor 104, co-processor circuitry 107, or an external system, needs to store a packet in memory 103, the entity reads a pointer from pointer cache 523 and uses the pointer to transfer the packet to external buffers 528 through SDRAM interface 522. Control logic 524 allocates the external buffer as the corresponding pointer is read from pointer cache 523. SDRAM stores the packet in the external buffer indicated by the pointer. Allocation means to reserve the buffer, so other entities do not improperly write to it while it is allocated.

When the entity no longer needs the external buffer—for example, the packet is transferred from memory 103 through SDRAM interface 522 to co-processor circuitry 107 or transmit interface 108—then the entity writes the pointer to pointer cache 523. Control logic 524 de-allocates the external buffer as the corresponding pointer is written to pointer cache 523. De-allocation means to release the buffer, so other entities may reserve it. The allocation and de-allocation process is repeated for other external buffers 528.

Bass et al., U.S. patent 6,804,249 discloses a system and method of moving information units from a network processor toward a data transmission network in a prioritized sequence which accommodates several different levels of service. The present invention includes a method and system for scheduling the egress of processed information units (or frames) from a network processing unit according to service based on minimum bandwidth specifications where position in the queue is adjusted after each service based on minimum bandwidth specification and the length of frame, a process which is subject to rounding errors. To avoid the accumulation of rounding errors inequitably influencing the position of some in the queue, a system to adjust for the rounding errors adds an increased measure of fairness to the system. The scheduler system illustrated in FIG. 3 is comprised of a plurality of flows 210, time-based calendars 220, 230, 250, weighted fair queueing (WFQ) calendars 240 and

target port queues 260.

Holt et al., U.S. patent 5,790,545 discloses a method and packet switch for efficient switching of a plurality of received packets from a plurality of ingress ports to a plurality of egress ports, using the steps of: A) storing the packets in memory; B) sending arrival information for each packet to a destination egress port for the packet; C) storing, in memory at each destination egress port, the arrival information; D) requesting, by each destination egress port, the packets from the packet memory in accordance with a predetermined scheme; and E) sending, by the packet memory, to the destination egress ports, the packets requested. At column 9, lines 25-55 states:

> First, where the EPs have initialized (1502) the ACTIVE bit to be equal to False for all connections, a received packet that arrives at an IP via an ingress port is stored (1504) in a queue in a memory, wherein each queue corresponds to an individual connection; Then, for each packet that arrives at the IP, the IP determines (1506) whether the packet has arrived at an empty queue, and where the packet has arrived at an empty queue, the IP generates (1508) an arrival tag that has a connection identifier code, and where selected, may also include scheduling information. Where the arrival tag is generated, the IP sends the arrival tag to a destination EP for the packet. The EP receives (1510) the arrival tag and records its arrival, typically by setting an ACTIVE bit in a connection table coupled to the EP, where the connection table has an ACTIVE bit for each connection that goes through the EP. The EP selects (1512), from among all the connections that have the ACTIVE bit set in the connection table, a connection in accordance with a predetermined scheme, clears the ACTIVE bit for the selected connection, and sends a request containing at least a connection identifier back to the IP that sent the arrival tag. Upon receiving the request, the IP uses (1514) the connection identifier to locate the packet queue and dequeues a next packet, labels the packet with a MORE bit which is True if there are more packets remaining in the queue and False otherwise, and sends the packet to the requesting EP. Then, the EP receives (1516) the requested packet and transmits the packet from the associated egress port. Where the MORE bit in the packet is True (1518), the EP also sets (1520) the ACTIVE bit for the connection.

Applicants respectfully submit that each of the independent claims 1, 12, and 19, as amended, is patentable over all the art of record. Independent claims 1, 12,

and 19, as amended, respectively define a scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array and a plurality of calendars for scheduling said flows, a QoS scheduler for implementing Quality-of-Service (QoS) scheduling for a plurality of flows, and a computer program product for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array and a plurality of calendars for scheduling said flows in a scheduler.

Known high-performance network processor scheduler systems are able to search entire calendars in one system cycle for the purpose of updating calendar status, that is active flow status. High-performance schedulers will no longer be able to search entire calendar arrays within one system cycle as performance requirements increase. Bandwidth constraints no longer allow the entire calendar array to be searched each cycle. The present invention as recited in independent claims 1, 12, and 19, as amended, use of the cache for storing a cache copy subset of said active flow indicators from said calendar status array (CSA). The cache copy data contained in the cache is used to determine if a given calendar is ready to dequeue a frame. The cache copy data contained in the cache is used with the CSA to allow simultaneous access to determine a flow of a calendar for servicing.

As amended, each of the independent claims 1, 12, and 19 further define the cache for storing a cache copy subset or the step of storing a subset of said active flow indicators from said calendar status array (CSA) in a cache utilizing a current pointer (CP) to a calendar entry and loading said subset of said active flow indicators

from said calendar status array (CSA) starting at said current pointer (CP) CSA entry. The use of the cache for storing a cache copy subset of said active flow indicators from said calendar status array (CSA) utilizing a current pointer (CP) to a calendar entry and loading said subset of said active flow indicators from said calendar status array (CSA) starting at said current pointer (CP) CSA entry, is not disclosed or suggested in the references of record including Henrion et al., Snyder II et al., Bass et al., and Holt et al.

Thus, each of the independent claims 1, 12, and 19, as amended, is patentable.

Dependent claims 2-11, 13-18 and 20 further define the invention of patentable independent claims 1, 12, and 19, as amended, and are likewise patentable.

Applicants have reviewed all the art of record, and respectfully submit that the claimed invention is patentable over all the art of record, including the references not relied upon by the Examiner for the rejection of the pending claims.

It is believed that the present application is now in condition for allowance and allowance of each of the pending claims 1-20, as amended is respectfully requested. Prompt and favorable reconsideration is respectfully requested.

If the Examiner upon considering this amendment should find that a telephone interview would be helpful in expediting allowance of the present application, the Examiner is respectfully urged to call the applicants' attorney at the number listed below.

Respectfully submitted,

By: Joan Pennington

Joan Pennington
Reg. No. 30,885
Telephone: (312) 670-0736